Case Study on Software Effort Estimation

Tülin Erçelebi Ayyıldız and Hasan Can Terzi

***Abstract*—Since most of the projects encounter effort overruns, effort estimation is one of the most important estimates of software projects during software development. There are several software effort estimation methodologies in the literature. However, instead of proposing a novel effort estimation methodology finding the necessary attributes that affects the software effort estimation is an also important contribution. This study focuses on analyzing the necessity of these attributes. We apply linear regression technique to investigate relation between these attributes. Lastly we evaluate our prediction performance with using Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE), MSE (Mean Square Error) and Prediction Quality (pred(e)). In order to conduct case study we used the Desharnais (77 projects) dataset from the publicly available PROMISE software engineering repository. Results show that the attribute "PointsNonAdjust" is the most necessary attribute in order to estimate software effort.**

***Index Terms*—Desharnais dataset, linear regression, software effort estimation.**

I. Introduction

Software effort estimation plays a vital role in a successful software project because it effects on the project's cost, duration, quality, performance. Because of this necessity there are many effort estimation methodologies in the literature [1], [2], [3].

However, instead of proposing a new methodology finding which attributes affect the software effort is also an important issue.

For this reason, we have focused on analyzing the importance of the attributes in software effort estimation. In order to conduct case study we used Desharnais dataset from the PROMISE software engineering repository. First of all, we analyze the correlation between each attributes of Desharnais dataset and effort attribute. We apply linear regression technique to investigate relation between these attributes. Lastly we evaluate our prediction performance with using Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE), MSE (Mean Square Error) and Prediction Quality (pred(e)).

In this study, we ask two research questions related to the Desharnais dataset:

1. What is the influence of each of the attributes on software effort estimation?

2. What is the influence of all of the attributes combined on software effort estimation?

Manuscript received February 10, 2017; revised May 5, 2017.

T. E. A. and H. C. T. are with the Computer Engineering Department, Başkent University, Ankara, Turkey (e-mail: ercelebi@ baskent.edu.tr, hasancanterzi@ gmail.com).

We have shown that some attributes are more necessary than other attributes.

The rest of the study is organized as follows: Section II summarizes the literature research. In Section III projects and datasets analyzed in the study are introduced, Section IV provides case studies, Section V provides prediction accuracies, finally the conclusions are given in Section VI.

II. Literature Summary

Several effort estimation measures have been defined so far. COCOMO is the one of the widely used software effort estimation methodology in the literature. It is an algorithmic software effort estimation methodology developed by Barry Boehm. This methodology also uses a linear regression formula together with parameters derived from historical projects and existing project specifications. There are three different levels of COCOMO which are Basic, Intermediate and Detailed. The basic COCOMO is used to make quick estimates for small and medium-sized projects. Effort Adjustment Factor (EAF) is used in the intermediate and detailed COCOMO methodology.

Wideband Delphi is another software effort estimation methodology. It is a consensus based effort estimation technique and effort is predicted based on the judgments of one or more expert(s) [4]. This methodology is suitable when the consultants are familiar with the projects to be developed. Sometimes, the methodology may fail to reach a consensus, and judgment errors might occur.

Use Case Point (UCP) methodology is also another software effort estimation methodology. UCP is the basic technique proposed by Gustav Karner [5] for estimating effort based on Use Cases. The method assigns quantitative weight factors (WF) to actors and use cases based on their classification as Simple, Average and Complex.

In order to measure the software size and estimate the required effort, unadjusted use case weight, unadjusted actor weight, technical complexity factors (TCF), environmental factors (EF) and productivity are taken into account.

Several approaches can be used to convert the size obtained from the use case point evaluation to the required effort. For example; Karner's methodology assumes the productivity of 20 person hours per adjusted UCP.

III. Analyzed Dataset

The Desharnais dataset [6] is composed of a total of 81 projects developed by a Canadian software house in 1989. Each project has twelve attributes which are described in Table I. The projects 38, 44, 65 and 75 contain missing attributes, so only 77 complete projects are used.

doi: 10.18178/ijiee.2017.7.3.670

TABLE I: ATTRIBUTE DEFINITION FOR DESHARNAIS DATASET

Attribute	Descriptions
Project	Project ID which starts by 1 and ends by 81
TeamExp	Team experience measured in years
Manager Exp	Manager experience measured in years
YearEnd	Year the project ended
Length	Duration of the project in months
Effort	ActualEffort is measured in person-hours
Transactions	Transactions is a count of basic logical transactions in the system
Entities	Entities is the number of entities in the systems data model
PointsAdj	Size of the project measured in unadjusted function points.
Envergure	Function point complexity adjustment factor.
PointsNonAdjust	Size of the project measured in adjusted function points.
Language	Type of language used in the project expressed as 1, 2 or 3.

The Desharnais dataset has become very popular as many developers use it in addition to other datasets to train and evaluate software estimation models.

This data set includes nine numerical attributes. The eight independent attribute of this data set, namely ‘‘TeamExp”, ‘‘ManagerExp”, ‘‘YearEnd”, ”Length”, ‘‘Transactions”, ‘‘Entities”, ‘‘PointsAdj”, ‘‘Envergure”, and ‘‘PointsNonAjust” are all considered for constructing the models. The dependent attribute ‘‘Effort” is measured in person hours.

In Desharnais dataset, The Project and the Language attributes are not considered in this study. Because Project attribute is Project ID starts by 1 and ends by 81. It does not make sense for our study. Moreover, Language attribute is categorical. Therefore, these two attributes ignored from the dataset.

IV. CASE STUDY

In this section, the correlations between attributes of Desharnais dataset and software effort are analyzed and applicability of the regression analysis is examined.

The correlation between two variables is a measure of how well the variables are related. The most common measure of correlation in statistics is the Pearson Correlation (or the Pearson Product Moment Correlation - PPMC) which shows the linear relationship between two variables. Pearson correlation coefficient analysis produces a result between -1 and 1. A result of -1 means that there is a perfect negative correlation between the two values at all, while a result of 1 means that there is a perfect positive correlation between the two variables. Results between 0.5 and 1.0 indicate high correlation.

The Pearson correlation coefficients between attributes and software efforts are given in Table II for Desharnais dataset.

Please note that, all of the statistical analyses in this study are performed by Minitab statistical tool [7].

As it can be seen from Table II, Length, Transactions, Entities, PointsAdjust and PointsNonAdjust attributes' correlation coefficients are above 0.50. Since correlation coefficient values are greater than 0.50 it means there is a strong correlation between dependent and independent variables. Moreover, these attributes p-values are also smaller than 0.05 threshold. So, we can conclude that these attributes are statistically significant [8].

TABLE II: PEARSONS'S CORRELATION COEFFICIENTS AND P-VALUES

Attribute	Effort	Correlation Coefficient	P-value
TeamExp	Effort	0.259	0.023
ManagerExp	Effort	0.160	0.164
YearEnd	Effort	-0.031	0.786
Length	**Effort**	**0.652**	**0.000**
Transactions	**Effort**	**0.500**	**0.000**
Entities	**Effort**	**0.583**	**0.000**
PointsAdjust	**Effort**	**0.703**	**0.000**
Envergure	Effort	0.417	0.000
PointsNonAdjust	**Effort**	**0.725**	**0.000**

In order to show the differences between the actual and estimated values of the dependent variable (obtained by applying the regression equation), scatterplots and residual plots are given for “PointsAdj” and “PointsNonAdjust” attributes (which have the highest correlation coefficients) in Fig. 1 through Fig. 4.

Scatterplots of the dependent (each attribute) and independent variables (effort) can be used to observe the linearity of the data points. In a scatterplot, the continuous line shows the regression line that represents the relationship between the dependent and independent variable. Data points correspond to dependent variable versus independent variable of the projects. Note that when the data points are close to regression line, the prediction accuracy is high.

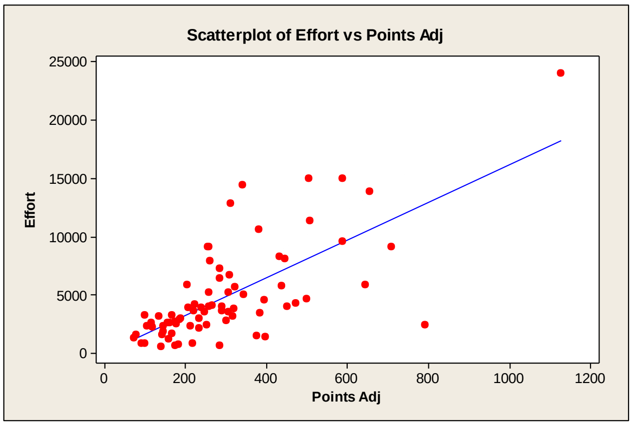

Fig. 1. Scatterplot of pointsadj vs. the effort.

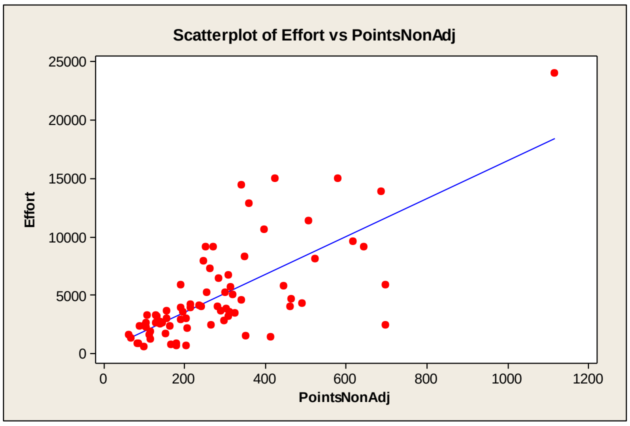

Fig. 2. Scatterplot of pointsnonadj vs. the effort.

According to scatterplots, most of the data points are closer to regression line. But there are some outliers that can be realized from the Fig. 1 and Fig. 2.

Residual is also a graph that shows the difference between the actual and estimated values of the dependent variable. The linear regression analysis said to be appropriate if the data points in a residual plot are randomly scattered in the graph; otherwise, a non-linear model would be more suitable [9].

The residual plots for "PointsAdj" and "PointsNonAdjust" attributes are given in Fig. 3 and Fig. 4.

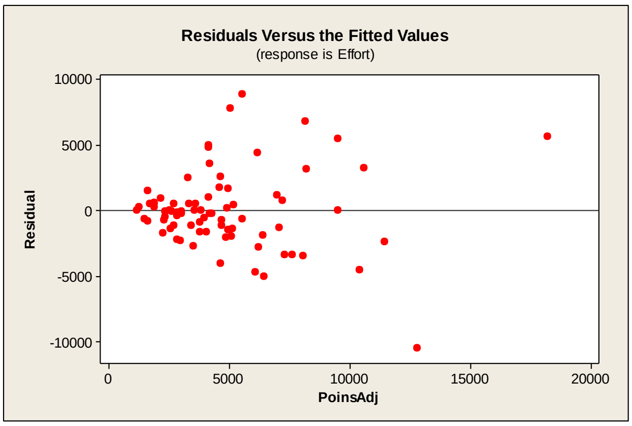

Fig. 3. The residuals vs. the pointsadj against the effort.

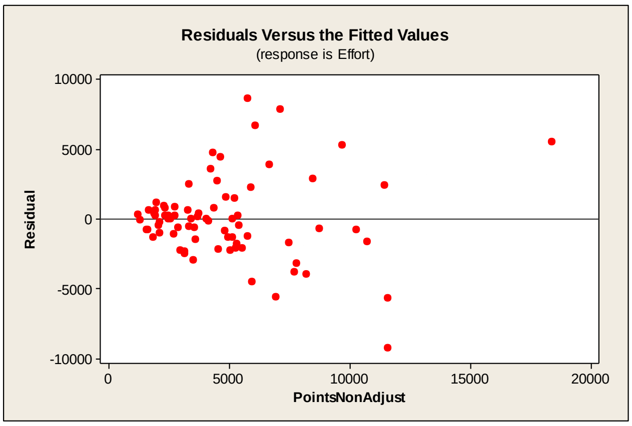

Fig. 4. The residuals vs. the pointsnonadj against the effort.

According to residual plots, data points in a residual plot are randomly scattered. Therefore after observing high correlation coefficients, obtaining statistical significant p values, given scatterplots and residual plots now we can apply linear regression analysis.

In Table III, Regression Equations and Coefficient of determination (R^2) values are given. The R^2 value is for example 42.60 means that 42.60% of the variation in Effort can be explained by the independent variables.

TABLE III: LINEAR REGRESSION EQUATIONS

Equations		R^2
Effort=282+403 Length	(1)	42.6
Effort=1867+16.7 Transactions	(2)	34.0
Effort=1901+24.3 Entities	(3)	25.0
Effort=15+16.2 PointsAdj	(4)	49.5
Effort=232+16.3 PointsNonAdjust	(5)	52.6

According to results, only the (5) give acceptable R^2 values (an acceptable value of R^2 is ≥ 0.5 [10]).

We also try to predict software effort with using all attributes (Length, Transactions, Entities, PointsAdj and PointsNonAdjust). We applied stepwise linear regression method. According to stepwise linear regression we obtain the following regression equation which is given in Table IV:

TABLE IV: STEPWISE LINEAR REGRESSION EQUATION

Effort=-230.13 + 34PointsNonAdjust + 213 Length - 19 PointsAdj - 6.8 Transactions	(6)	R^2=59.44

V. PREDICTION ACCURACY

As evaluation criteria, we used Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE), Prediction Quality (Pred (e)) and Mean Squared Error (MSE).

Prediction quality (Pred (e) = k/n) is calculated on a set of n projects, where k is the number of projects for which MRE is less than or equal to "e", where "e" is the selected threshold value for MRE. The interpretation of MRE and Pred criteria is that the accuracy of an estimation technique is proportional to the Pred and inversely proportional to the MRE.

Finally, MSE is the statistical measure of the average squares of the errors. Two or more models can be compared by using their MSEs as a measure of how well they explain a given set of observations. The smaller MSE values are the better.

In this study, we compute prediction quality for e=0.30. Tate and Verner suggested that for an acceptable estimation model the value of Pred (0.30) should exceed 0.70 [11].

According to Hastings and Sajeev's evaluation. if MMRE and MdMRE values are between 0.20 and 0.50, it is considered as acceptable [12]. MRE, MMRE and MdMRE are defined as:

$$MRE = \frac{|ActualValue - EstimatedValue|}{ActualValue} \quad (7)$$

$$MMRE = \frac{1}{n} + \sum_{i=1}^{n} MRE_i \quad (8)$$

$$MdMRE = \text{median}(MRE_i) \quad (9)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(AVi - EVi)^2 \quad (10)$$

where *AVi* is the actual value, *EVi* is the estimated value of the ith project and n is the number of projects.

Prediction Accuracy results according to given equations are given in Table V.

TABLE V: PREDICTION ACCURACY RESULTS

Project ID	**MRE**					
	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
Project 1	0.007	0.182	0.386	0.038	0.001	0.014
Project 2	0.664	0.085	0.128	0.074	0.048	0.309
Project 3	0.149	2.149	3.173	1.031	0.969	0.214
Project 4	0.400	0.360	0.252	0.354	0.350	0,030
Project 5	0.119	0.957	0.948	0.771	0.686	0.068
Project 6	0.329	0.236	0.441	0.073	0.192	0.048
Project 7	0.522	0.500	0.137	0.021	0.010	0.070
Project 8	0.411	0.271	0.191	0.011	0.049	0.029
Project 9	0.348	0.397	0.486	0.462	0.458	0.412
Project 10	0.218	0.309	0.166	0.218	0.211	0.426
Project 11	1.150	0.145	0.059	0.063	0.007	0.503
Project 12	0.212	0.524	0.489	0.537	0.486	0.259
Project 13	0.347	0.060	0.600	0.248	0.270	0.482
Project 14	0.160	0.096	0.189	0.130	0.100	0.166
Project 15	0.215	0.123	0.027	0.123	0.095	0.093

Project 16	1.168	1.384	0.897	0.682	0.676	0.645
Project 17	0.098	0.117	0.077	0.488	0.376	0.105
Project 18	0.724	0.126	1.787	0.814	0.614	0.768
Project 19	0.318	0.551	1.510	0.427	0.285	0.501
Project 20	1.735	2.376	2.247	0.792	0.945	0.924
Project 21	0.658	0.521	0.436	0.364	0.352	0.415
Project 22	0.455	0.356	0.164	0.190	0.153	0.308
Project 23	0.602	0.208	0.115	0.231	0.302	0.025
Project 24	0.211	0.344	0.641	0.414	0.366	0.224
Project 25	0.120	0.158	0.697	0.179	0.216	0.255
Project 26	0.872	0.044	1.367	0.623	0.670	1.119
Project 27	0.238	0.138	1.149	0.404	0.501	0.513
Project 28	0.385	0.014	1.285	0.791	0.925	1.215
Project 29	0.072	0.475	0.168	0.359	0.377	0.187
Project 30	0.501	0.213	1.186	0.859	0.962	1.190
Project 31	0.312	0.189	0.276	0.142	0.152	0.226
Project 32	4.506	5.040	2.978	3.197	3.184	3.135
Project 33	0.609	0.965	0.563	0.687	0.887	0.974
Project 34	0.641	0.203	0.358	0.277	0.238	0.408
Project 35	1.290	5.100	3.749	3.378	3.863	2.605
Project 36	0.219	0.545	0.475	0.544	0.523	0.322
Project 37	2.286	3.664	1.815	3.266	3.149	2.428
Project 38	1.236	4.319	2.937	3.168	2.738	0.823
Project 39	0.164	0.142	0.326	0.099	0.088	0.294
Project 40	0.154	1.034	0.336	0.753	0.687	0.064
Project 41	4.946	4.487	1.172	4.468	3.936	3.754
Project 42	0.983	0.350	0.478	0.114	0.099	0.300
Project 43	0.125	0.268	0.259	0.253	0.216	0.043
Project 44	0.255	0.683	0.335	0.453	0.523	0.388
Project 45	0.024	0.517	0.483	0.052	0.039	0.189
Project 46	0.632	0.684	0.511	0.606	0.523	0.408
Project 47	0.504	0.560	0.222	0.014	0.180	0.356
Project 48	0.328	0.403	0.222	0.429	0.422	0.366
Project 49	0.038	0.273	0.221	0.024	0.072	0.263
Project 50	0.139	0.053	0.014	0.298	0.245	0.250
Project 51	0.506	0.615	0.665	0.615	0.598	0.514
Project 52	0.981	0.117	0.639	0.010	0.090	0.296
Project 53	0.403	0.116	0.114	0.014	0.244	0.315
Project 54	0.329	0.682	0.373	0.148	0.279	0.280
Project 55	0.562	0.150	0.411	0.170	0.287	0.034
Project 56	0.288	0.446	0.162	0.053	0.083	0.261
Project 57	0.338	0.115	0.429	0.152	0.008	0.412
Project 58	0.714	0.684	0.701	0.178	0.211	0.152
Project 59	0.749	0.357	0.538	0.295	0.221	0.371
Project 60	0.439	0.894	0.596	0.269	0.140	0.166
Project 61	0.088	0.258	0.709	0.144	0.002	0.325
Project 62	2.444	0.884	1.308	0.455	0.294	0.993
Project 63	0.033	0.408	0.174	0.304	0.372	0.246
Project 64	0.443	0.361	0.439	0.276	0.252	0.263
Project 65	2.403	1.027	1.095	0.055	0.045	0.977
Project 66	0.099	0.218	0.156	0.250	0.237	0.341
Project 67	2.733	2.077	1.845	1.229	0.852	0.895
Project 68	3.945	5.386	4.351	3.152	2.380	1.039
Project 69	1.427	0.804	0.658	0.509	0.284	0.419
Project 70	0.182	0.090	0.185	0.264	0.181	0.153
Project 71	3.538	8.116	5.176	6.812	4.951	0.794
Project 72	0.282	0.295	0.544	0.233	0.174	0.090
Project 73	2.656	3.065	3.291	3.616	3.986	4.202
Project 74	0.828	1.021	0.312	0.741	0.812	0.850
Project 75	0.046	0.111	0.308	0.002	0.081	0.261
Project 76	0.130	0.650	0.051	0.780	0.972	0.799
Project 77	0.382	0.304	0.676	0.237	0.230	0.644
Pred(0.30)	**0.34**	**0.40**	**0.30**	**0.50**	**0.52**	**0.44**
MMRE	**0.77**	**0.93**	**0.84**	**0.71**	**0.67**	**0.59**
MdMRE	**0.40**	**0.36**	**0.47**	**0.30**	**0.28**	**0.32**

As the results indicate all of the regression models give predictive MdMRE results according to Hastings and Sajeev's evaluation. According to MMRE, the best result belongs to (6) which is 0.59. Although, there is no acceptable prediction quality result, 0.50 and 0.52 values are the most closest the acceptance level.

For an overall decision we should examine MSE results. The MSE results for all equations are given in Table VI.

TABLE VI: MSE RESULTS

Equations	MSE
1	9934823
2	11426025
3	12980969
4	8740096
5	8211092
6	9649301

According to MSE results, since the smaller MSE values are the better, the prediction models can be best constructed with equations (5), (4), (6), (1), (2) and (3), respectively.

We can conclude that, for Desharnais dataset, effort can be best predicted by "PointsNonAdjust" attribute. This attribute is size of the Project measured in unadjusted function points and it is calculated as Transactions plus Entities.

$$PointsNonAdjust = Transactions + Entities \quad (11)$$

The second attribute is "PointsAdj" and it is size of the Project masured in adjusted function points. This is calculated as:

$$PointsAdj = PointsNon\ Adjust \ x\ (0.65 + 0.01xEnvergure) \quad (12)$$

Although, adjustment factors are used to reduce the % deviation of the estimated value from the actual value, it is not relevant for this dataset.

VI. CONCLUSION

This study focused on the finding the most necessary attributes which are required for software effort estimation. We used Desharnais dataset's attributes. Among the twelve attributes in the Desharnais dataset, five attributes were selected according to Pearson correlation coefficients and p-values. These include "Length", "Transactions", "Entities", "PointsAdj" and "PointsNonAdjust". The effort estimation analysis was carried out based on a linear regression model. The first five equations obtained with using each attribute as an independent variable. Equation (6) is obtained with using stepwise linear regression model. In this model, the necessary attributes were selected among five attributes. The prediction accuracy evaluation conducted was based on four criteria which include MMRE, MdMRE, PRED(0.30) and MSE. The results obtained showed that the attribute "PointsNonAdjust" is the most necessary attribute. The next necessary attribute is "PointsAdj", followed by stepwise linear regression result and the next necessary attributes are "Length", "Transactions" and "Entities", respectively.

This study showed that, for the Desharnais dataset, the effort can be best predicted by using "PointsNonAdjust" attribute. Hence we can conclude that, there is no need to use adjustment factor in order to estimate software effort for Desharnais dataset.

REFERENCES

[1] A. Živković, I. Rozman, and M. Heričko, "Automated software size estimation based on function points using UML models," *Information and Software Technology*, vol. 47, no. 13, pp. 881-890, 2005.

[2] T. E. Ayyıldız and A. Koçyiğit, "Correlations between problem domain and solution domain size measures for open source software,"

40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), pp.81-84, 2014.
[3] T. E. Ayyıldız and A. Koçyiğit, "An early software effort estimation method based on use cases and conceptual classes", *Journal of Software*, vol. 9, no. 8, pp. 2169-2173, 2014.
[4] J. Anderson, E. Branch, T. Luedtke, S. Carson, J. Falconi, and R. Janda, "Parametric estimating handbook: Reinvention laboratory," *DOD, NASA*, 1999.
[5] G. Karner, "Metrics for objector," *Diploma thesis, University of Linkoping*, Sweden, 1993.
[6] J. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des function," *Master's Thesis, University of Montreal*, 1989.
[7] *Minitab 17 Statistical Software, State College,* PA: Minitab, Inc. (www.minitab.com), 2010.
[8] B. Ozkan, O. Turetken, and O. Demirörs, "Software functional size: For cost estimation and more," *Software Process Improvement Communications in Computer and Information Science*, vol. 16, pp. 59-69, 2008.
[9] J. Miles, "Residual plot, wiley statsref: Statistics reference online," 2014.
[10] C. DeSanto, M. Totoro, and R. Moscartelli, "Introduction to statistics 9th edition," *Pearson,* 2010.
[11] G. Tate and J. Verner, "Software costing in practice," *The Economics of Information and Software*, R. Veryard, Butterworth-Heinemann, pp. 101-126, 1990
[12] T. E. Hastings and A. S. M. Sajeev, "A vector based approach to software size measurement and effort estimation," *IEEE Transactions on Software Engineering*, vol. 24, no. 4, pp. 337-350, 2001.

Tülin Erçelebi Ayyıldız received B.Sc degree in computer engineering department from Çankaya University, Turkey, in 2005 and M.Sc. degree in computer engineering department from Hacettepe University, Turkey, in 2008. She received Ph.D degree in Information systems department from Middle East Technical University (METU), Turkey, in 2015. She is now an instructor at Başkent University in the department of computer engineering.

Dr. Erçelebi Ayyıldız's main research interests are software size and effort estimation, software measurement and software process improvement.

Hasan Can Terzi received B.Sc degree in management information systems from Başkent University, Turkey, in 2012 and M.Sc. degree in computer engineering department from Başkent University, Turkey, in 2017. He is consulting CMMI model and ISO 9000 standards on the companies. His main research interests are software quality, software process improvement and software size and effort estimation.